# Optimizing Preventive Maintenance Models

Michael Bartholomew-Biggs
School of Physics Astronomy and Mathematics, University of Hertfordshire

Bruce Christianson
School of Computer Science, University of Hertfordshire

Ming Zuo
Department of Mechanical Engineering, University of Alberta

## Presented at ICCOPT1, August 2004

Part of this work was carried out while the first author was a Visiting Professor in the Department of Mechanical Engineering at the University of Alberta in December 2003.

**Preventive maintenance**
Oiling the wheels is
almost as effective as
turning the clock back

## 1 Scheduling preventive maintenance (PM)

The following ideas are due to Lin, Zuo & Yam

• Frequency of system failure depends on its *age*.

Number of failures between $t = a$ and $t = b$ is

$$\int_a^b h(t)dt$$

where $h(t)$ is the *hazard rate function* .

• PM makes system's *effective age* $<$ calendar age.

A system enters service at time $t = 0$

First PM is at time $t_1 = x_1$.

Just *before* PM, effective age $y_1 =$ calendar age $x_1$.

Just *after* PM, effective age is $b_1 x_1$, for some $b_1 < 1$.

From $t_1$ till next PM at $t = t_2$, effective age is

$$y = b_1 x_1 + x, \quad \text{where} \quad 0 < x < t_2 - t_1.$$

Failure rate after PM may not be same as a *genuinely* younger system.

- number of failures between $t = 0$ and $t = t_2$ is

$$\int_0^{x_1} h(x)dx + \int_0^{x_2} a_1 h(b_1 x_1 + x)dx.$$

Here $x_2 = t_2 - t_1$ and $a_1$ is a constant $\geq 1$

The effective age just before PM at time $t_2$ is

$$y_2 = b_1 x_1 + x_2$$

PM reduces this to $b_2 y_2$, where $b_1 \leq b_2 \leq 1$.

Thus, between $t_2$ and $t_3$,

effective age is

$$y = b_2 y_2 + x = b_2 b_1 x_1 + b_2 x_2 + x,$$

where $0 < x < x_3 = t_3 - t_2$;

and number of failures is

$$\int_0^{x_3} a_2 a_1 h(b_2 y_2 + x)dx$$

for some $a_2 \geq 1$.

Generalising, for $k = 1, .., n$,

$y_k$ = effective age just before $k$-th PM at time $t_k$.
$x_k = t_k - t_{k-1}$, the $k$-th PM interval

This implies

$$t_k = \sum_{i=1}^{k} x_i \tag{1.1}$$

$$y_k = b_{k-1} y_{k-1} + x_k = (\sum_{j=1}^{k-1} B_j x_j) + x_k \tag{1.2}$$

where $B_j = \Pi_{i=j}^{k-1} b_i$.

$$x_k = y_k - b_{k-1} y_{k-1}. \tag{1.3}$$

*Cumulative hazard rate*

$$H_k(t) = \int A_k h(t) dt \quad \text{where} \quad A_k = \Pi_{i=1}^{k-1} a_i.$$

Number of failures between $t_{k-1}$ and $t_k$ is

$$H_k(y_k) - H_k(b_{k-1} y_{k-1}).$$

Now suppose PM takes place $n-1$ times
- the $n$-th PM is a system replacement.

For an optimal PM schedule we minimize

$$C(y) = \frac{R_c}{T}$$

$$= \frac{\gamma_r + (n-1) + \gamma_m \sum_{k=1}^{n} [H_k(y_k) - H_k(b_{k-1} y_{k-1})]}{y_n + \sum_{k=1}^{n-1} (1 - b_k) y_k} \qquad (1.4)$$

where

$$\gamma_r = \frac{\text{Cost of system replacement}}{\text{Cost of PM}}$$

$$\gamma_m = \frac{\text{Cost of minimal system repair}}{\text{Cost of PM}}$$

$R_c$ reflects lifetime cost (multiple of one PM cost)

$T$ is the total life of the system

Hence $C(y)$ is *mean cost* of operating the system.

Lin, Zuo & Yam have proposed a semi-analytic method for finding $y_k$ to minimize (1.4).

Their approach also optimizes $n$, the number of PM

They quote results when hazard rates are Weibull functions

$$h(t) = \beta t^{\alpha-1} \quad \text{with } \beta > 0 \text{ and } \alpha > 1 \tag{1.5}$$

We use numerical methods to minimize mean cost
- initially we get optimum $n$ by explicit enumeration.

We need to avoid $y_k < 0$
- so introduce transformation $y_k = u_k^2$ and minimize

$$\tilde{C}(u) = \frac{\gamma_r + (n-1) + \gamma_m \sum_{k=1}^{n}[H_k(u_k^2) - H_k(b_{k-1}u_{k-1}^2)]}{u_n^2 + \sum_{k=1}^{n}(1-b_k)u_k^2}.$$

We consider example hazard rates of the form

$$h(t) = \beta_1 t^{\alpha-1} + \beta_2; \quad \text{with } \beta_1,\ \beta_2 > 0 \text{ and } \alpha > 1, \tag{1.6}$$

for various choices of $\alpha, \beta_1$ and $\beta_2$.

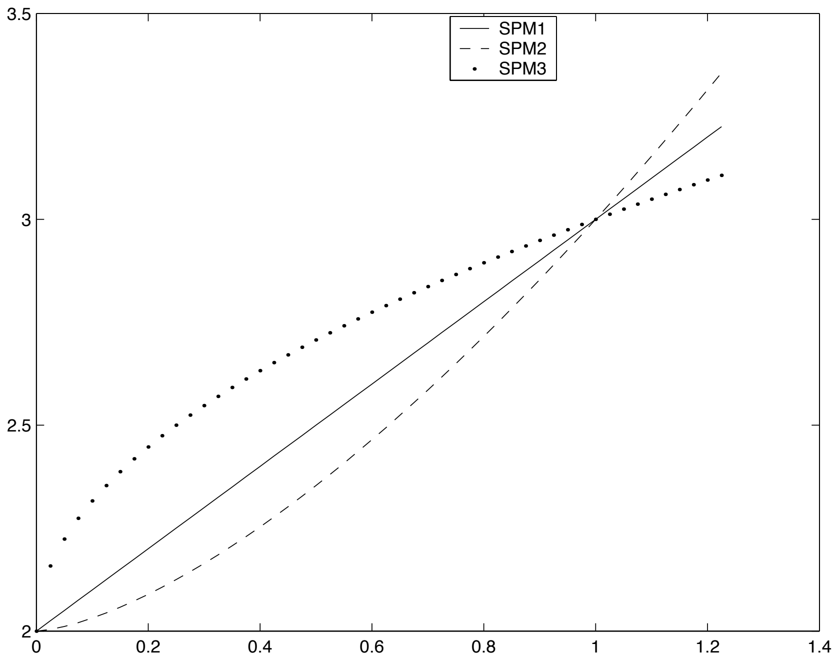


Figure 1: Sample hazard-rate functions $h(t)$

We use cost ratios

$$\gamma_m = 10 \ \text{ and } \ \gamma_r = 1000 \tag{1.7}$$

=> system much more expensive to replace than to repair or maintain.

$\tilde{C}(u)$ minimized by Newton's method for fixed $n$

$\nabla\tilde{C}(u)$ and $\nabla^2\tilde{C}(u)$ obtained via fortran90 AD module `oprad` (Brown, Christianson)
- reverse accumulation approach for AD
- interface with `oprad` simplifies coding of changes to PM model

Solution of **SPM1** when $n = 7$.

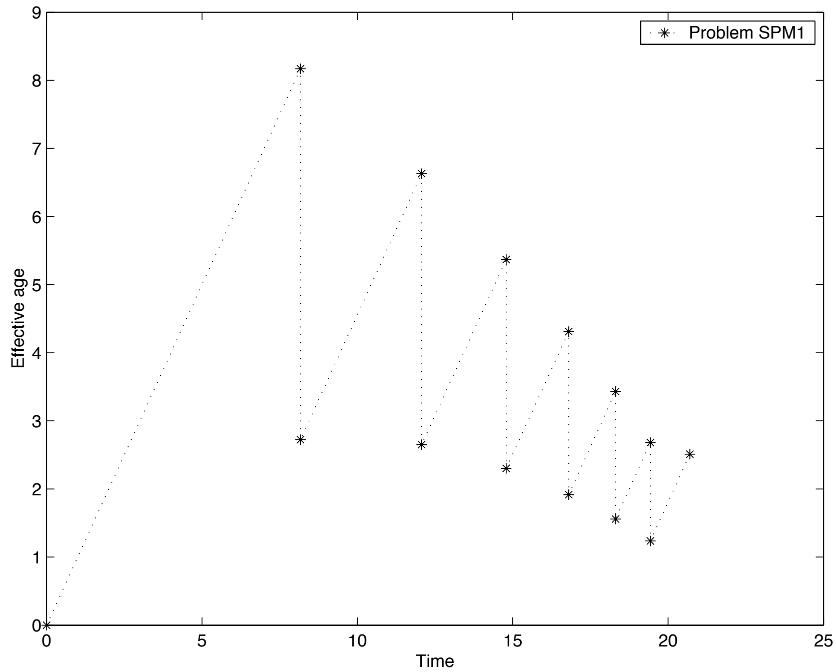


Figure 2: Optimal solution to **SPM1** for $n = 7$

- plots effective age against time
- instantaneous decrease every time PM occurs.
- system becomes *effectively younger* at each PM.
- intervals between PM get shorter

Newton iterations show that $\tilde{C}(u)$ is non-convex
- function may have several local minima.

There are *trivial* multiple solutions due to $y = u^2$

To test for multiple *distinct* solutions, we applied the global method DIRECT (Jones) to $\tilde{C}$.

DIRECT is derivative-free and seeks global minimum in hyperbox defined by bounds on variables.
- systematically subdivides initial box
- only explores *potentially optimal* regions

After obtaining a solution $u_1^*, .., u_n^*$ (e.g. by Newton's method) we use DIRECT in the box

$$0 \le u_i \le 2\bar{u} \quad \text{where} \quad \bar{u} = \frac{1}{n}\sum_{i=1}^{n} u_i^*.$$

To date we have not found better minimum of $\tilde{C}$
- suggests Newton's method is indeed finding the global minimum of mean cost for each $n$.

## 2 Minimizing mean cost for varying $n$

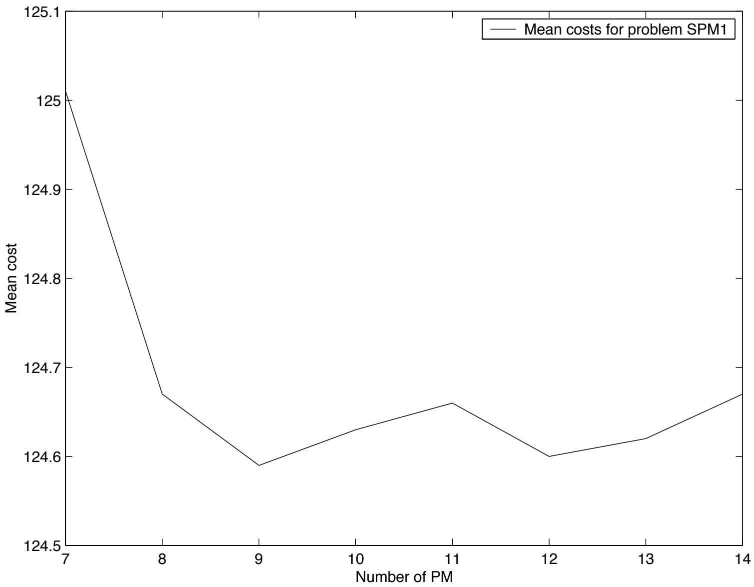


Figure 3: Solutions of **SPM1** for various $n$

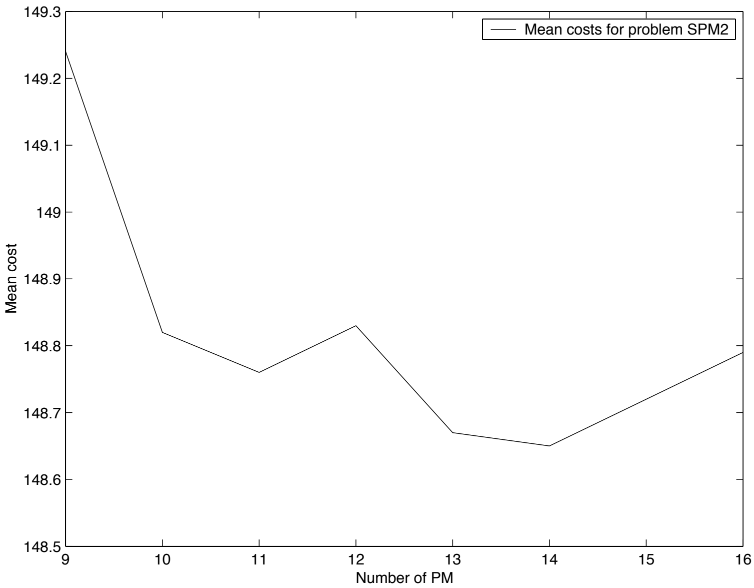


Figure 4: Solutions of **SPM2** for various $n$

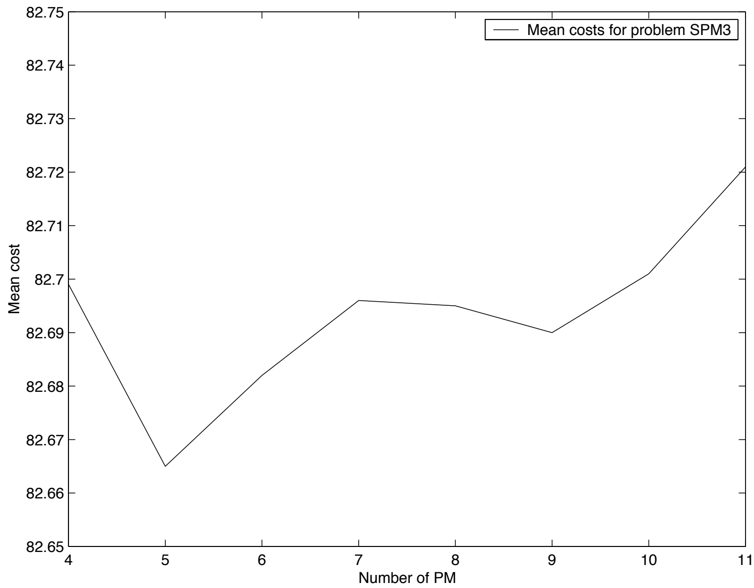


Figure 5: Solutions of **SPM3** for various $n$

In each graph the minimum with larger $n$ is spurious

- optimal effective-ages imply negative PM intervals!

It is better to optimize w.r.t. PM intervals:
Let $v_1, .., v_n$ be optimization variables and set

$$y_1 = x_1 = v_1^2 \tag{2.1}$$

and, for $k = 2, \ldots, n$,

$$x_k = v_k^2; \quad y_k = b_{k-1} y_{k-1} + x_k. \tag{2.2}$$

This ensures the $x$'s and $y$'s are all non-negative.

Now

$$\bar{C}(v) = C(y) \tag{2.3}$$

where $C(y)$ is mean cost function (1.4)

We can minimize $\bar{C}(v)$ by Newton method & `oprad`

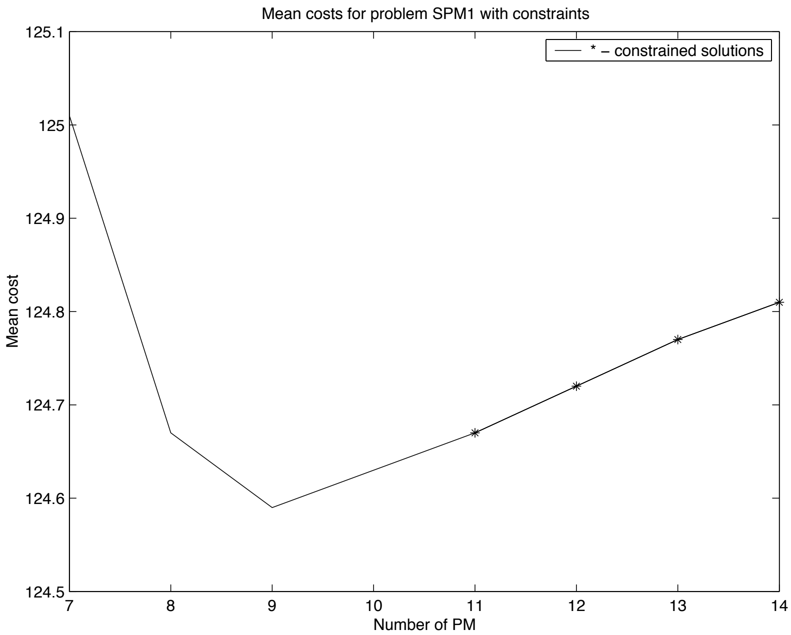


Figure 6: Solutions of **SPM1** using $\bar{C}(v)$ for various $n$

## 3 Minimizing mean cost w.r.t. $n$

We want to find the optimum number of PM without explicit enumeration.

Use continuous variable $\nu$ for number of PMs

Let $n$ denote the integer part of $\nu$ and set $\theta = \nu - n$.
- obviously $\theta < 1$ (but $\theta$ may be $\approx 1$).

There are $n-1$ *complete* PMs and one *partial* PM

Partial maintenance reduces effective age to

$$y_n - \theta(y_n - b_n y_n) = (1 - \theta + \theta b_n) y_n = \tilde{b}_n y_n$$

instead of $b_n y_n$.

There is a system replacement at effective age $y_{n+1}$
- (relative) cost of repairs between $t_{n-1}$ and $t_{n+1}$ is

$$\gamma_m [H_n(y_n) - H_n(b_{n-1} y_{n-1}) + H_{n+1}(y_{n+1}) - H_{n+1}(\tilde{b}_n y_n)].$$

Time elapsed between $t_{n-1}$ and $t_{n+1}$ is

$$y_n - b_{n-1} y_{n-1} + y_{n+1} - \tilde{b}_n y_n.$$

Let $N$ be the maximum number of PMs

We need optimization variables $y_1,\ldots,y_N$ and $\nu$.

Now perform the following calculations.

$$n = \lfloor \nu \rfloor; \quad \theta = \nu - n; \;\; \tilde{b}_n = 1 - \theta + \theta b_n \tag{3.1}$$

$$\begin{aligned} R_c = \gamma_r + (\nu - 1) + \gamma_m \sum_{k=1}^{n} [H_k(y_k) - H_k(b_{k-1} y_{k-1})] \\ + \gamma_m [H_{n+1}(y_{n+1}) - H_{n+1}(\tilde{b}_n y_n)] \end{aligned} \tag{3.2}$$

$$T = y_n + \sum_{k=1}^{n-1} (1 - b_k) y_k + y_{n+1} - \tilde{b}_n y_n. \tag{3.3}$$

$$C(y,\nu) = \frac{R_c}{T}. \tag{3.4}$$

$C(y,\nu)$ is continuous *but non-differentiable*
- there are jumps in derivatives because

$$\frac{\partial C}{\partial y_k} = 0 \;\text{ for }\; \nu < k-1; \qquad \frac{\partial C}{\partial y_k} \neq 0 \;\text{ when }\; \nu \geq k-1.$$

We want to minimize $C(y,\nu)$ subject to the constraint that PM intervals are non-negative
- therefore we require

$$y_k - b_{b-1}y_{k-1} \geq 0 \ \text{ for } \ k = 1,..,n-1 \tag{3.5}$$

$$\text{and } \ y_{n+1} - \tilde{b}_n y_n \geq 0. \tag{3.6}$$

This means the number of constraints depends on $\nu$.

We also want $\nu$ to be an integer and so

$$\theta(1-\theta) = 0. \tag{3.7}$$

Minimizing (3.4) subject to (3.5), (3.6), (3.7)
- use non-differentiable exact penalty function

$$C(y,\nu) + \rho_1 \sum_{k=2}^{n} |(y_k - b_{k-1}y_{k-1})_-|$$

$$+\rho_1|(y_{n+1} - \tilde{b}_n y_n)_-| + \rho_2|\theta(1-\theta)|. \tag{3.8}$$

where $(z)_-$ denotes $\mathrm{Min}(0,z)$.

Better to use PM intervals as variables
- we extend $\bar{C}(v)$ to include the extra variable $\nu$.

We calculate $\bar{C}$ by first setting

$$x_1 = v_1^2; \;\; y_1 = x_1;$$

and then, for $k = 2, .., n$,

$$x_k = v_k^2; \quad y_k = b_{k-1} y_{k-1} + x_k.$$

We then use (3.1) – (3.3) and finally set

$$\bar{C}(v, \nu) = \frac{R_c}{T}. \tag{3.9}$$

Scheduling problem is to minimize (3.9) subject only to the equality constraint (3.7).

This can be solved by minimizing

$$\bar{C}(v, \nu) + \rho_2 |\theta(1 - \theta)|. \tag{3.10}$$

We can seek (global) minimum of (3.10) by DIRECT.
- global because $\rho_2|\theta(1-\theta)|$ may produce multiple local minima when $\theta \approx 0$ or $\theta \approx 1$.

A semi-heuristic approach, based on restarts

**Algorithm A**

Choose a range $n_{min} \leq n \leq N$

Choose starting values $\hat{v}_k$, $k = 1, \ldots, N$.

Set starting value

$$\hat{\nu} = \frac{n_{min} + N}{2}.$$

Choose box-size $\pm\Delta v_k,\ \pm\Delta\nu$, for DIRECT as

$$\Delta v_k = 0.99\hat{v}_k, \quad k = 1, .., N; \quad \Delta\nu = \frac{N - n_{min}}{2}.$$

After $M$ iterations of DIRECT perform a *restart*
- search re-centred on $(v_k^*, \nu^*)$ – best point so far.

Box-size is reset to

$$\Delta v_k = Max(1, 0.99v_k^*), \quad k = 1, .., N;$$

$$\Delta\nu = Min(\nu^* - n_{min}, N - \nu^*)$$

Re-starts continue until $M$ DIRECT iterations give change $< 0.01\%$ in the value of $\bar{C}$.

Algorithm A was applied to **SPM1** – **SPM3** with

$$n_{min} = 1, \;\; N = 20, \;\; M = 100$$

Starting guesses

$$\hat{v}_1 = 5, \;\; \hat{v}_k = Max(0.9\hat{v}_{k-1}, 1), \;\; k = 2, .., N$$

Penalty parameter in (3.10) was $\rho_2 = 0.1$.

Results

| | $\bar{C}$ | Number of PM | DIRECT iterations | Restarts |
|---|---|---|---|---|
| **SPM1** | 124.59 | 9 | 400 | 3 |
| **SPM2** | 148.76 | 11 | 500 | 4 |
| **SPM3** | 82.665 | 5 | 300 | 2 |

Table 1: Scheduling solutions with Algorithm A

These optima agree with results from minimizing $\tilde{C}$ by Newton's method for fixed values of $n$.

Sensitivity of solutions to changes in repair and replacement cost

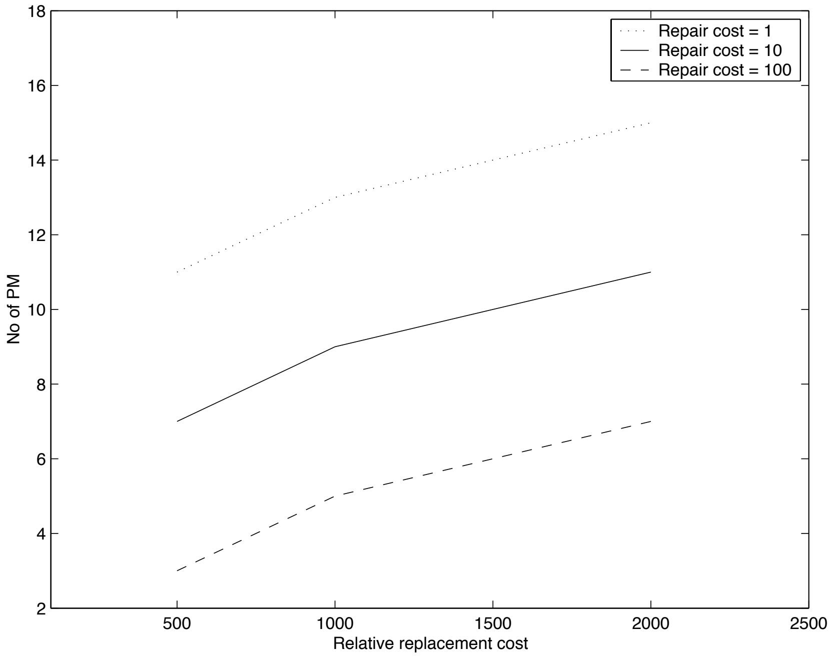


Figure 7: Solutions of **SPM1** for various $\gamma_m$, $\gamma_r$

Optimal $n$ increases as the repair cost comes closer to PM cost.

Conversely, optimal $n$ decreases as relative cost of repair increases.

Optimal $n$ increases and decreases with $\gamma_r$.

## 4 A differentiable alternative to (3.10)

*Fletcher's ideal penalty function* solves

$$\text{Minimize} \quad F(x) \quad \text{s.t.} \;\; c_i(x) = 0, \; i = 1, .., m$$

by unconstrained minimization of

$$E(x) = F - c^T (AA^T)^{-1} Ag + \rho c^T c \qquad (4.1)$$

where $g = \nabla F(x)$

$A$ is the Jacobian matrix whose rows are $\nabla c_k(x)^T$ for $k = 1, .., m$.

It would be good to use this in Algorithm A
- instead of the non-smooth penalty function

We could then refine DIRECT estimates of the global solution by using a gradient-based method

Another change in formulation is needed ...

$N$ is the largest number of PM permitted

Optimization variables are effective ages $y_1, .., y_N$
- together with extra quantities $\theta_1, .., \theta_N$.

The $\theta_k$ lie between 0 and 1
- to indicate if $k$-th PM is complete or partial.

$k$-th PM reduces effective age from $y_k$ to $\tilde{b}_k y_k$ where

$$\tilde{b}_k = 1 - \theta_k + \theta_k b_k.$$

Hence repair cost between $t_k$ and $t_{k+1}$ is

$$\gamma_m [H_{k+1}(y_{k+1}) - H_{k+1}(\tilde{b}_k y_k)].$$

Total cost of all PMs is

$$\sum_{k=1}^{N-1} \theta_k$$

So lifetime cost of the system is

$$R_c = \gamma_r + \sum_{k=1}^{N-1} \theta_k + \gamma_m [\sum_{k=1}^{N-1} H_{k+1}(y_{k+1}) - H_{k+1}(\tilde{b}_k y_k)].$$

Life of the system is

$$T = y_N + \sum_{k=1}^{N-1} (1 - \tilde{b}_k) y_k.$$

$R_c$ and $T$ are defined in terms of $y_1, ..., y_N, \theta_1, ..., \theta_N$ and are differentiable. Hence cost function

$$\tilde{C}(y, \theta) = \frac{R_c}{T} \tag{4.2}$$

is also differentiable.

We need to minimize $\tilde{C}(y,\theta)$ subject to

$$\theta_k(1-\theta_k) = 0, \quad k = 1,..,N \tag{4.3}$$

(so no partial PMs in an optimum schedule)

Clearly (4.3) is differentiable.

Minimizing (4.2) subject to (4.3) can be expected to produce a solution where for some $n \le N$

$$\theta_k = 1, \quad k = 1,..,n;$$

$$\theta_k = 0, \quad y_k = y_{k-1}, \quad k = n+1,..,N.$$

For the problem of minimizing (4.2) subject to (4.3) the ideal penalty function turns out to be

$$E(y,\theta) = C(y,\theta) - \sum_{k=1}^{N} \frac{\theta_k(1-\theta_k)}{1-2\theta_k} \frac{\partial C}{\partial \theta_k} + \rho \sum_{k=1}^{N} \theta_k^2(1-\theta_k)^2.$$

This is differentiable and its global minimum gives an optimal PM schedule.

Global minimum can be estimated by DIRECT and refined by a fast local gradient method.

## 5 Conclusions

• We can do PM scheduling via numerical methods as well as analytical approach of Lin, Zuo and Yam.
- may be important when hazard rates are not simple

• Use of AD makes it easy to implement changes in problem formulation.

• Can treat number of PMs as a continuous variable.
- **Algorithm A** applies *global* minimization to a non-smooth function. Gives promising results.

• A variant of Algorithm A could use Fletcher's ideal penalty function $E(y,\theta)$
- permits solution refinement by a gradient method.

Even though $E(y,\theta)$ involves $\nabla C(y,\theta)$
- and so $\nabla^2 C(y,\theta)$ is involved in $\nabla E(y,\theta)$ -
$\nabla E$ can be obtained using AD (Christianson)
- implementation remains a topic for further work